AFRL-RB-WP-TR-2009-3034

NONLINEAR STRUCTURES OPTIMIZATION FOR FLEXIBLE FLAPPING WING MAVS

Robert L. Walker Jr. and Craig E. Svanberg

Design and Analysis Methods Branch
Structures Division

FEBRUARY 2009
Final Report

Approved for public release; distribution unlimited.

See additional restrictions described on inside pages

STINFO COPY

AIR FORCE RESEARCH LABORATORY
AIR VEHICLES DIRECTORATE
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7542
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE

NOTICE AND SIGNATURE PAGE

AFRL-RB-WP-TR-2009-3034 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

*//Signature//

ROBERT L. WALKER JR
Research Aerospace Engineer
Design and Analysis Methods Branch
Structures Division

//Signature//

DENIS P. MROZINSKI
Chief
Design and Analysis Methods Branch
Structures Division

//Signature//

DAVID M. PRATT
Technical Advisor
Structures Division
Air Vehicles Directorate

*Disseminated copies will show "//Signature//" stamped or typed above the signature blocks.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YY)*
February 2009

2. REPORT TYPE
Final

3. DATES COVERED *(From - To)*
01 April 2007 – 30 September 2008

4. TITLE AND SUBTITLE
NONLINEAR STRUCTURES OPTIMIZATION FOR FLEXIBLE FLAPPING WING MAVS

5a. CONTRACT NUMBER
In-house

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
0602201

6. AUTHOR(S)
Robert L. Walker Jr. (AFRL/RBSD)
Craig E. Svanberg (AFRL/AFIT)

5d. PROJECT NUMBER
A0F0

5e. TASK NUMBER

5f. WORK UNIT NUMBER
0A

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Design and Analysis Methods Branch (AFRL/RBSD)
Structures Division, Air Force Research Laboratory
Air Vehicles Directorate
Wright-Patterson Air Force Base, OH 45433-7542
Air Force Materiel Command, United States Air Force

Air Force Institute of Technology (AFRL/AFIT)

8. PERFORMING ORGANIZATION REPORT NUMBER
AFRL-RB-WP-TR-2009-3034

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory
Air Vehicles Directorate
Wright-Patterson Air Force Base, OH 45433-7542
Air Force Materiel Command
United States Air Force

10. SPONSORING/MONITORING AGENCY ACRONYM(S)
AFRL/RBSD

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)
AFRL-RB-WP-TR-2009-3034

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES
PAO Case Number: 88ABW-2009-0483, 17 Feb 2009. Report contains color.
This is a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT
This report documents the culmination of in-house and AFIT collaboration in the area of nonlinear structures optimization for flexible flapping wing MAVs. The work was divided into four different tasks, including: Preliminary design of flapping wing device, Couple Abaqus with aerodynamic model, Comparison of experimental and computational results, and Evaluate CSIRF progress for transition to FY09 6.2 program. Technology developed in this task is being transitioned in the VAES program A0GI0A "MAV Hover Flight Sciences."

15. SUBJECT TERMS
nonlinear optimization, flapping wing, fluid structure interaction, micro-air vehicles, flexible wing, flapping mechanism

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT:	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON (Monitor)
Unclassified	Unclassified	Unclassified	SAR	22	Robert L. Walker

19b. TELEPHONE NUMBER *(Include Area Code)*
N/A

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18

The design, experimentation, construction, and simulation of Flapping Micro Air Vehicles are relatively new areas of research being studied in many universities, industries, and military organizations. For universities in particular, flapping MAVs provide an area of research that is easy to enter into. Apart from the many grants and financing offered, this area of research is affordable in both start up and ongoing materials cost. Most of the materials to get started in the design and construction of flapping MAVs at an entry level are readily available and can be purchased at a local hoppy shop. Because of this, there are many emerging studies however; very few have the more advanced capability to collect data computationally and experimentally. That is where the Air Force Research Laboratory/Air Vehicles Directorate can excel. They plan to become a center of excellence in the world for MAVs. This project is a step in that direction.

The project “Nonlinear Structures Optimization for Flexible Flapping Wing Micro-Air Vehicles” was funded with Chief Scientist Innovative Research funds. This project was divided into 4 parts, Preliminary design of flapping wing device, Couple Abaqus with aerodynamic model, Comparison of experimental and computational results, and Evaluate CSIRF progress for transition to FY09 6.2 program. These four parts are discussed below.

Part 1

Milestone 1: Preliminary design of flapping wing device, measuring wing position and structural deformations, measuring 6 DOF forces and moments

As part of this project, a Masters student, Lt Craig Svanberg, focused his thesis on the proof of concept and development of a flapping wing device, various testing and data collection techniques for these small scale flapping wings [1]. The preliminary design of a flapping wing device was accomplished in the 4th quarter of FY07. The wing’s position and structural deformations were able to be captured via high speed camera system; however no scale has been set to capture the exact values. Only 1 DOF, the normal force for the flapping mechanism, was able to be accurately captured. The other 5 DOFs were too noisy with interference, however, future research has been planned to accurately capture the remaining 5 DOFs. This study has increased both AFIT’s and AFRL’s understanding and capabilities for future research. Lt Svanberg’s Thesis, “Biomimetic Micro Air Vehicle Testing Development and Small Scale Flapping-Wing Analysis,” can be seen as Attachment A: Report Number AFIT/GAE/ENY/08-M27.

Part 2

Milestone 2: Couple Abaqus with aerodynamic model as a computational tool modeling flapping wing flight through temporal sub-iterations or through cyclic iterations

Apart from physical experiments in a laboratory setting, computer simulations can also provide a wealth of data and knowledge. An accurate computer simulation can not only give one the results of an experimental lab set up, but also make available a viable method for optimization.

Abaqus is a high-fidelity general-purpose finite element capability that AFRL/RB uses. This software is capable of linear or non-linear, implicit or explicit, and dynamic or static solutions. Abaqus however does not incorporate aerodynamic loads in its solution. A general static or dynamic solution does not require aerodynamic loads in order to be accurate. However, for a computational simulation of flapping wing flight to be precise, aerodynamic loads will need to be present to solve for the correct forces, moments, deformations, lift, drag and thrust applied to the modeled wings.

For the aero code, ultimately a high-fidelity code that can handle low Reynolds numbers would be idea. AFRL/RB is currently working on low, medium and high fidelity codes that can handle these low Reynolds numbers. At the time of this project, a fully completed aero code was not available. So while Abaqus was not able to be fully coupled with an true aerodynamic model, proof of capability for Abaqus to be coupled with an aero model was verified still able to be verified. The capability for Abaqus to be coupled with an external aerodynamics code was examined using Abaqus to build and run the FEM model, an estimated Aero Code to predict a 10% resisting load to the model, and Python Scripting to wrap around everything.

Building the Model in Abaqus CAE

The flapping wing in the model consisted of a 0.02 inch diameter rod, 2.5 inches long attached to an actuator at its root. The actuator was set to a periodic flapping of 10 Hz. The simulation was only taken out through 1 period of time, t = 0.1 seconds with a peak total amplitude of 114 degrees. This Abaqus/aero code "solution" was developed through cyclic iterations rather than temporal sub-iterations.

Create an Input file

By opening Abaqus CAE, the graphical user interface, a model of the whole system or job can be created. This includes the wing structure, mesh, elements, boundary conditions, actuations, loads, output requests, and time steps. Once the model is developed in Abaqus CAE, an Input file can be written. The input file can be opened and edited in Notepad; WireRod.inp can be seen in Attachment B.

Create a grid

In the input file, the grid is made up of node, x, y, and z positions. For the Rod part it looked as follows:

*Part, name=WireLine

*Node	X	Y	Z
1,	0.,	0.,	0.
2,	0.5,	0.,	0.
3,	1.,	0.,	0.
4,	1.5,	0.,	0.
5,	2.,	0.,	0.
6,	2.5,	0.,	0.

Create a kinematics file

Abaqus needs a kinematics equation or tabulated data entry for the root of the wing to actuate as desired. In this case, tabulated data was used for a 1-dimensional 10 Hz sin wave for one period t = 0.0-0.1 seconds. There are 51 discrete time values in this example.

*Amplitude, name=sin10tab (time, amplitude)

0., 0., 0.002, 0.125333, 0.004, 0.24869, 0.006, 0.368125, 0.008, 0.481754, 0.01, 0.587785, 0.012, 0.684547, 0.014, 0.770513, 0.016, 0.844328, 0.018, 0.904827, 0.02, 0.951057, 0.022, 0.982287 0.024, 0.998027, 0.026, 0.998027, 0.028, 0.982287, 0.03, 0.951057, 0.032, 0.904827, 0.034, 0.844328, 0.036, 0.770513, 0.038, 0.684547, 0.04, 0.587785, 0.042, 0.481754, 0.044, 0.368125, 0.046, 0.24869 0.048, 0.125333, 0.05, 1.22515e-16, 0.052, -0.125333, 0.054, -0.24869, 0.056, -0.368125, 0.058, -0.481754, 0.06, -0.587785, 0.062, -0.684547, 0.064, -0.770513, 0.066, -0.844328, 0.068, -0.904827, 0.07, -0.951057, 0.072, -0.982287, 0.074, -0.998027, 0.076, -0.998027, 0.078, -0.982287, 0.08, -0.951057, 0.082, -0.904827, 0.084, -0.844328, 0.086, -0.770513, 0.088, -0.684547, 0.09, -0.587785, 0.092, -0.481754, 0.094, -0.368125, 0.096, -0.24869, 0.098, -0.125333, 0.1, -2.4503e-16

Create a Forces-at-Nodes file

In this example there are 6 nodes. Node 1 is at the root of the wing, and will carry no forces. Nodes 2-6 will carry a force for each of the 51 discrete times, t. Tabulated data in the form of 51 x (time, force) for each node is created. Likewise, moments can be created at each node and time.

Node 2: 51 x (time, force)
Node 3: 51 x (time, force)
Node 4: 51 x (time, force)
Node 5: 51 x (time, force)
Node 6: 51 x (time, force)

Start Loop:

Start Python Script

A python script can hold multiple command line prompts.

Run Input file

Run the input file (WireRod.inp): Abaqus input=WireRod

Write U, V and A to file

The python script that writes one or all of displacements, velocities, accelerations, rotational displacements, rotational velocities, and rotational accelerations, (U, V, A, UR, VR, and AR) is called UVAatNodes.py and can be seen in Attachment C. A, UR, VR, and AR were commented out in this case using the # symbol. Notice that only U and V were written to the file UVA.txt in Attachment D.

```
tStep-1
Node =   1 U[x] = 0.00000000, U[y] = 0.00000000, u[z] = 0.00000000
Node =   2 U[x] = 0.00000000, U[y] = 0.00000000, u[z] = 0.00000000
Node =   3 U[x] = 0.00000000, U[y] = 0.00000000, u[z] = 0.00000000
Node =   4 U[x] = 0.00000000, U[y] = 0.00000000, u[z] = 0.00000000
Node =   5 U[x] = 0.00000000, U[y] = 0.00000000, u[z] = 0.00000000
Node =   6 U[x] = 0.00000000, U[y] = 0.00000000, u[z] = 0.00000000
tStep-2
Node =   1 U[x] = 0.00000000, U[y] = 0.00000000, u[z] = 0.00000000
Node =   2 U[x] = -0.00010812, U[y] = 0.01039741, u[z] = 0.00000000
Node =   3 U[x] = -0.00015341, U[y] = 0.01712761, u[z] = 0.00000000
Node =   4 U[x] = -0.00016988, U[y] = 0.02118577, u[z] = 0.00000000
Node =   5 U[x] = -0.00017518, U[y] = 0.02348901, u[z] = 0.00000000
Node =   6 U[x] = -0.00017691, U[y] = 0.02480394, u[z] = 0.00000000
```

Aero code reads U, V, A
Aero code writes Forces-at-Nodes

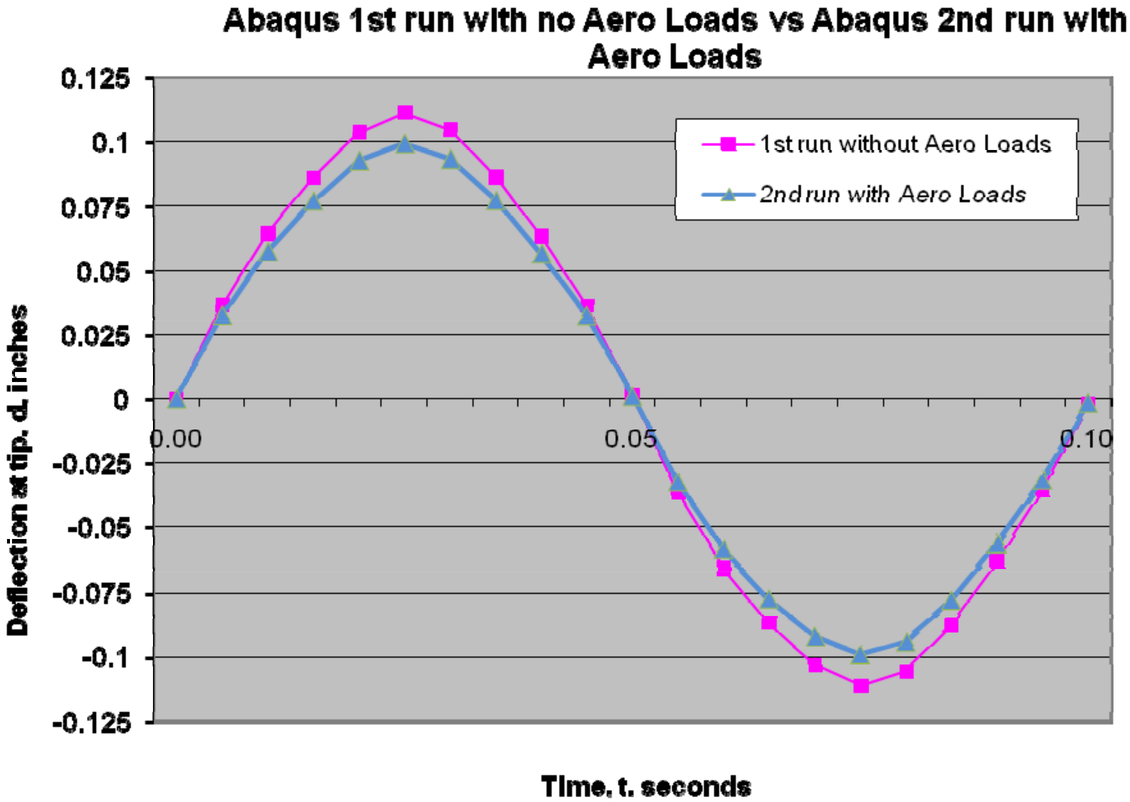
The aero code reads in the position, velocity, and acceleration of each node at each time step finds its solution and in turn outputs a file of Node, Force in x direction, Force in y direction, and Force in z direction (Fx, Fy, and Fz) for each of the 51 time steps. These values of forces and moments at each node are all Abaqus will need to update the next iteration. An estimation of an approximately 10% resisting force was applied in the opposite direction of the velocity at each time step and node. DLOAD.txt can be seen in Attachment E.

```
tStep-1
Node =   1 F[x] = 0.00000000, F[y] = 0.00000000, F[z] = 0.00000000
Node =   2 F[x] = 0.00000000, F[y] = 0.00000000, F[z] = 0.00000000
Node =   3 F[x] = 0.00000000, F[y] = 0.00000000, F[z] = 0.00000000
Node =   4 F[x] = 0.00000000, F[y] = 0.00000000, F[z] = 0.00000000
Node =   5 F[x] = 0.00000000, F[y] = 0.00000000, F[z] = 0.00000000
Node =   6 F[x] = 0.00000000, F[y] = 0.00000000, F[z] = 0.00000000
tStep-2
Node =   1 F[x] = 0.00000000, F[y] = 0.00000000, F[z] = 0.00000000
Node =   2 F[x] = -0.00578963, F[y] = 0.08462580, F[z] = 0.00000004
Node =   3 F[x] = -0.00936150, F[y] = 0.15126581, F[z] = 0.00000089
Node =   4 F[x] = -0.01221042, F[y] = 0.21087037, F[z] = 0.00000190
Node =   5 F[x] = -0.01508493, F[y] = 0.27074319, F[z] = 0.00000251
Node =   6 F[x] = -0.00230485, F[y] = 0.03090382, F[z] = -0.00000002
```

Python Script updates Forces-at-Nodes values in the Rod-Input file.
End Loop when the solution converges on itself.

Convergence Verified

Results from the first and second run of the loop for Node #6 are shown below. After the 10% resisting force is applied to the Abaqus model it is clearly seen that the solution changes. When the new Abaqus solutions are fed into the aero code, the newer forces were less in magnitude than the initial forces. As a result the 3rd runs solution in this simple case will fall in between the previous two solutions. This example converged very well after only 5 or 6 runs.



From this, it can be said that once a viable aero code is available, coupling Abaqus with the code using the previous method will be attainable. The CPU processing time for a true aero code such as a Vortex Lattice method solver will take more than 95% of the processing time for any convergence.

Part 3
Milestone 3: Comparison of experimental results and computational results complete

The experimental flapping results do not include any exact position or deformations to be compared to computational results. Any computational results would include position and deformations at any node desired. However the Abaqus results would not include aero loads as access to a low-Reynolds number medium fidelity aero code was not available as previously thought. Two and three dimensional vortex lattice codes are currently being worked on in-house at AFRL/RB. There will be plenty of opportunities in the future to verify and validate results of both experimental and computational results.

Part 4
Milestone 4: Evaluate CSIRF progress for transition to FY09 6.2 program

The work in this project has already led to a myriad of opportunities for improvement and transition to future 6.2 work. Studies are currently ongoing with the next year group of AFIT students to obtain all 6 DOF accurately, capture precise position and deformation values, and develop more advanced flapping wing devices. Whether or not testing can be done in a vacuum chamber is also being looked into. If this is probable, then results can be compared directly to an Abaqus model alone. Also, once an aero code becomes available, it should be relatively straightforward to couple the two, as Abaqus has shown that it is able to be linked external system. This project was successfully transitioned to the 6.2 project "Hovering Flight Sciences."

References
[1] C.E. Svanberg, "Biomimetic Micro air Vehicle Testing Development and Small Scale Flapping-Wing Analysis," March 2008.

Attachment B

```
*Heading
** Job name: LineIa Model name: FlappingLine
*Preprint, echo=NO, model=NO, history=NO, contact=NO
**
** PARTS
**
*Part, name=WireLine
*Node
      1,           0.,           0.,           0.
      2,          0.5,           0.,           0.
      3,           1.,           0.,           0.
      4,          1.5,           0.,           0.
      5,           2.,           0.,           0.
      6,          2.5,           0.,           0.
*Element, type=B31
1, 1, 2
2, 2, 3
3, 3, 4
4, 4, 5
5, 5, 6
*Nset, nset=_PickedSet2, internal
 1, 2
*Elset, elset=_PickedSet2, internal
 1,
*Nset, nset=_PickedSet5, internal
 1, 2
*Elset, elset=_PickedSet5, internal
 1,
*Nset, nset=_PickedSet6, internal, generate
 1,  6,  1
*Elset, elset=_PickedSet6, internal, generate
 1,  5,  1
** Region: (CircleSection:Picked), (Beam Orientation:Picked)
*Elset, elset=_I1, internal
 1,
** Section: CircleSection  Profile: circle
*Beam General Section, elset=_I1, poisson = 0.3, density=0.00017,
section=CIRC
0.01
0.,0.,-1.
 2e+07, 1.2e+07
*Damping, alpha=3.
** Region: (CircleSection:Picked)
*Elset, elset=_I2, internal, generate
 2,  5,  1
** Section: CircleSection  Profile: circle
*Beam General Section, elset=_I2, poisson = 0.3, density=0.00017,
section=CIRC
0.01
0.,0.,-1.
 2e+07, 1.2e+07
*Damping, alpha=3.
```

```
*End Part
**
**
** ASSEMBLY
**
*Assembly, name=Assembly
**
*Instance, name=WireLine-1, part=WireLine
*End Instance
**
*Element, type=CONN3D2
1,  , WireLine-1.1
*Connector Section, elset=_PickedSet6
Join, Rotation
*Nset, nset=Wire-1-Set-1, instance=WireLine-1
 1,
*Elset, elset=Wire-1-Set-1
 1,
*Elset, elset=_PickedSet6, internal
 1,
*Elset, elset=_PickedSet8, internal
 1,
*Elset, elset=_PickedSet9, internal
 1,
*Elset, elset=_PickedSet11, internal
 1,
*Elset, elset=_PickedSet12, internal
 1,
*Elset, elset=_PickedSet13, internal
 1,
*Nset, nset=_PickedSet14, internal, instance=WireLine-1
 2,
*Nset, nset=_PickedSet15, internal, instance=WireLine-1
 3,
*Nset, nset=_PickedSet16, internal, instance=WireLine-1
 4,
*Nset, nset=_PickedSet17, internal, instance=WireLine-1
 5,
*Nset, nset=_PickedSet18, internal, instance=WireLine-1
 6,
*End Assembly
*Amplitude, name=n1
0., 0., 0.002, 5.83239e-08, 0.004, 1.00638e-07, 0.006, 5.60611e-08
0.008, 5.28955e-08, 0.01, 7.8951e-08, 0.012, 3.42409e-08, 0.014, 6.64435e-
09
0.016, 3.74796e-08, 0.018, 2.91513e-08, 0.02, 2.18218e-08, 0.022,
2.59273e-08
0.024, 2.47149e-08, 0.026, -4.12702e-08, 0.028, -2.72973e-08, 0.03,
5.08839e-10
0.032, -2.338e-08, 0.034, -2.87364e-08, 0.036, -1.84142e-08, 0.038, -
5.95595e-08
0.04, -8.32315e-08, 0.042, -4.55654e-08, 0.044, -4.71681e-08, 0.046, -
7.38792e-08
```

0.048, -4.5457e-08, 0.05, -3.46535e-08, 0.052, -8.61659e-08, 0.054, -
8.78447e-08
0.056, -4.52237e-08, 0.058, -5.47121e-08, 0.06, -6.18404e-08, 0.062, -
2.02594e-08
0.064, -2.01634e-08, 0.066, -5.44641e-08, 0.068, -3.73712e-08, 0.07, -
1.7278e-08
0.072, -2.74173e-08, 0.074, 8.40318e-10, 0.076, 2.93238e-08, 0.078,
6.1483e-10
0.08, -6.60815e-09, 0.082, 3.32018e-08, 0.084, 3.76589e-08, 0.086,
3.25197e-08
0.088, 6.67889e-08, 0.09, 6.71171e-08, 0.092, 2.93088e-08, 0.094,
4.49846e-08
0.096, 7.15876e-08, 0.098, 5.1235e-08, 0.1, 5.83141e-08
*Amplitude, name=n2
0., 0., 0.002, 1.265e-07, 0.004, 1.98816e-07, 0.006, 7.85534e-08
0.008, 1.18093e-07, 0.01, 2.02067e-07, 0.012, 5.16148e-08, 0.014, -
2.48024e-08
0.016, 9.56447e-08, 0.018, 7.13413e-08, 0.02, -1.3886e-09, 0.022,
9.16473e-08
0.024, 7.56824e-08, 0.026, -1.16451e-07, 0.028, -5.65906e-08, 0.03,
3.93204e-08
0.032, -6.17116e-08, 0.034, -7.75568e-08, 0.036, -2.59821e-09, 0.038, -
1.19512e-07
0.04, -2.0557e-07, 0.042, -7.7836e-08, 0.044, -7.32229e-08, 0.046, -
1.71753e-07
0.048, -8.46376e-08, 0.05, -4.10294e-08, 0.052, -1.95801e-07, 0.054, -
1.93055e-07
0.056, -6.53403e-08, 0.058, -1.19391e-07, 0.06, -1.45296e-07, 0.062, -
7.62328e-09
0.064, -2.82e-08, 0.066, -1.44708e-07, 0.068, -6.79898e-08, 0.07, -
1.10964e-08
0.072, -8.35958e-08, 0.074, -6.40802e-09, 0.076, 9.25917e-08, 0.078, -
1.52145e-08
0.08, -4.45598e-08, 0.082, 9.12338e-08, 0.084, 8.48347e-08, 0.086,
3.71343e-08
0.088, 1.50064e-07, 0.09, 1.60074e-07, 0.092, 2.94592e-08, 0.094,
7.91209e-08
0.096, 1.68906e-07, 0.098, 8.85805e-08, 0.1, 1.01554e-07
*Amplitude, name=n3
0., 0., 0.002, 2.21858e-07, 0.004, 3.09074e-07, 0.006, 6.46338e-08
0.008, 1.6045e-07, 0.01, 3.24999e-07, 0.012, 5.69442e-08, 0.014, -
2.39885e-08
0.016, 2.10052e-07, 0.018, 1.00388e-07, 0.02, -8.01637e-08, 0.022,
1.55226e-07
0.024, 1.31244e-07, 0.026, -1.82612e-07, 0.028, -6.06676e-08, 0.03,
1.01833e-07
0.032, -1.35001e-07, 0.034, -1.68979e-07, 0.036, 3.12797e-08, 0.038, -
1.55522e-07
0.04, -3.27676e-07, 0.042, -9.31111e-08, 0.044, -8.78651e-08, 0.046, -
3.00499e-07
0.048, -1.45466e-07, 0.05, -4.37745e-08, 0.052, -3.04522e-07, 0.054, -
2.79682e-07
0.056, -5.43113e-08, 0.058, -1.94451e-07, 0.06, -2.55138e-07, 0.062,
5.38153e-09

0.064, -4.92144e-08, 0.066, -2.53992e-07, 0.068, -6.52349e-08, 0.07, 3.49325e-08
0.072, -1.57326e-07, 0.074, -3.56408e-08, 0.076, 1.67005e-07, 0.078, -3.27626e-08
0.08, -8.18388e-08, 0.082, 1.82141e-07, 0.084, 1.36893e-07, 0.086, 2.04098e-09
0.088, 2.20086e-07, 0.09, 2.63886e-07, 0.092, 2.55907e-08, 0.094, 1.2675e-07
0.096, 2.98441e-07, 0.098, 1.18898e-07, 0.1, 1.19129e-07
*Amplitude, name=n4
0., 0., 0.002, 3.25901e-07, 0.004, 4.36289e-07, 0.006, 3.29171e-08
0.008, 1.74755e-07, 0.01, 4.42952e-07, 0.012, 6.34649e-08, 0.014, -1.59914e-08
0.016, 3.66981e-07, 0.018, 1.31274e-07, 0.02, -1.69768e-07, 0.022, 1.72577e-07
0.024, 1.75828e-07, 0.026, -2.3649e-07, 0.028, -5.21735e-08, 0.03, 1.61458e-07
0.032, -2.39518e-07, 0.034, -2.85247e-07, 0.036, 7.04064e-08, 0.038, -1.7988e-07
0.04, -4.31022e-07, 0.042, -7.66451e-08, 0.044, -1.03855e-07, 0.046, -4.71805e-07
0.048, -2.28747e-07, 0.05, -3.88744e-08, 0.052, -4.11647e-07, 0.054, -3.50078e-07
0.056, -4.63906e-09, 0.058, -2.65654e-07, 0.06, -4.00696e-07, 0.062, -4.9351e-09
0.064, -7.58266e-08, 0.066, -3.64689e-07, 0.068, -4.02302e-08, 0.07, 1.11461e-07
0.072, -2.31552e-07, 0.074, -7.74686e-08, 0.076, 2.40842e-07, 0.078, -3.72107e-08
0.08, -9.98661e-08, 0.082, 2.96739e-07, 0.084, 1.84686e-07, 0.086, -6.40358e-08
0.088, 2.73873e-07, 0.09, 3.64889e-07, 0.092, 2.30882e-08, 0.094, 2.0002e-07
0.096, 4.60319e-07, 0.098, 1.41243e-07, 0.1, 1.07836e-07
*Amplitude, name=n5
0., 0., 0.002, 4.71521e-07, 0.004, 5.89313e-07, 0.006, -5.84963e-08
0.008, 1.23528e-07, 0.01, 5.91595e-07, 0.012, 1.28318e-07, 0.014, -1.34961e-08
0.016, 4.95595e-07, 0.018, 2.00777e-07, 0.02, -2.65703e-07, 0.022, 1.75302e-07
0.024, 2.07432e-07, 0.026, -2.87356e-07, 0.028, -4.96732e-08, 0.03, 2.07247e-07
0.032, -3.59772e-07, 0.034, -4.03405e-07, 0.036, 1.10279e-07, 0.038, -2.08935e-07
0.04, -5.20654e-07, 0.042, -3.16689e-08, 0.044, -1.24041e-07, 0.046, -6.7539e-07
0.048, -3.23593e-07, 0.05, -2.32369e-08, 0.052, -5.20613e-07, 0.054, -4.1615e-07
0.056, 7.284e-08, 0.058, -3.28414e-07, 0.06, -5.74307e-07, 0.062, -3.59483e-08
0.064, -9.62563e-08, 0.066, -4.70248e-07, 0.068, -1.25438e-08, 0.07, 2.00936e-07
0.072, -2.98303e-07, 0.074, -1.1934e-07, 0.076, 3.12152e-07, 0.078, -2.83845e-08

```
0.08, -1.01626e-07, 0.082, 4.20262e-07, 0.084, 2.25662e-07, 0.086, -
1.43304e-07
0.088, 3.21114e-07, 0.09, 4.59739e-07, 0.092, 1.98534e-08, 0.094,
2.91769e-07
0.096, 6.4188e-07, 0.098, 1.56801e-07, 0.1, 7.82015e-08
*Amplitude, name=sin10, definition=PERIODIC
1, 62.8319, 0., 0.
0., 1.
*Amplitude, name=sin10tab
0., 0., 0.002, 0.125333, 0.004, 0.24869, 0.006, 0.368125
0.008, 0.481754, 0.01, 0.587785, 0.012, 0.684547, 0.014, 0.770513
0.016, 0.844328, 0.018, 0.904827, 0.02, 0.951057, 0.022, 0.982287
0.024, 0.998027, 0.026, 0.998027, 0.028, 0.982287, 0.03, 0.951057
0.032, 0.904827, 0.034, 0.844328, 0.036, 0.770513, 0.038, 0.684547
0.04, 0.587785, 0.042, 0.481754, 0.044, 0.368125, 0.046, 0.24869
0.048, 0.125333, 0.05, 1.22515e-16, 0.052, -0.125333, 0.054, -0.24869
0.056, -0.368125, 0.058, -0.481754, 0.06, -0.587785, 0.062, -0.684547
0.064, -0.770513, 0.066, -0.844328, 0.068, -0.904827, 0.07, -0.951057
0.072, -0.982287, 0.074, -0.998027, 0.076, -0.998027, 0.078, -0.982287
0.08, -0.951057, 0.082, -0.904827, 0.084, -0.844328, 0.086, -0.770513
0.088, -0.684547, 0.09, -0.587785, 0.092, -0.481754, 0.094, -0.368125
0.096, -0.24869, 0.098, -0.125333, 0.1, -2.4503e-16
** ----------------------------------------------------------------
**
** STEP: Step-1
**
*Step, name=Step-1, nlgeom=YES, inc=100000
*Dynamic,alpha=-0.05,haftol=0.1
2e-06,0.1,1e-07,0.002
**
** BOUNDARY CONDITIONS
**
** Name: BC-1 Type: Connector displacement
*Connector Motion
_PickedSet11, 4
** Name: BC-2 Type: Connector displacement
*Connector Motion
_PickedSet12, 5
** Name: BC-3 Type: Connector displacement
*Connector Motion, amplitude=sin10tab
_PickedSet13, 6, 0.2
**
** LOADS
**
** Name: L1   Type: Concentrated force
*Cload, amplitude=n1
_PickedSet14, 2, -150
** Name: L2   Type: Concentrated force
*Cload, amplitude=n2
_PickedSet15, 2, -150
** Name: L3   Type: Concentrated force
*Cload, amplitude=n3
_PickedSet16, 2, -150
** Name: L4   Type: Concentrated force
```

```
*Cload, amplitude=n4
_PickedSet17, 2, -150
** Name: L5   Type: Concentrated force
*Cload, amplitude=n5
_PickedSet18, 2, -150
**
** OUTPUT REQUESTS
**
*Restart, write, frequency=0
**
** FIELD OUTPUT: F-Output-1
**
*Output, field, time interval=0.002, time marks=NO
*Node Output
A, U, V
**
** HISTORY OUTPUT: H-Output-1
**
*Output, history, variable=PRESELECT, time interval=0.005, time marks=NO
*End Step
```

Attachment C

```
from odbAccess import *
from abaqusConstants import *

odb = openOdb(path='1.odb')

filename = "UVA.txt"

print "writing to file %s" % filename

file = open(filename, 'w')

lastFrame = odb.steps['Step-1'].frames[-1]

for fieldName in lastFrame.fieldOutputs.keys():
    file.write(fieldName + "\n")

totaltime = .1
totalsteps = 50
itertime = totaltime/totalsteps

##################
#U displacement at nodes 1-6

for i  in range(51):

      lastFrame = odb.steps['Step-1'].frames[i]

      iteration = "iteration %3.2d" %(i)
      for stepName in odb.steps.keys():
#         file.write(stepName + "\n")
#         file.write(iteration + "\n")
#         timex = itertime*i
#         time = "Time = %6.6f" % (timex)
          file.write("\n")
#         file.write(time + "\n")

      displacement=lastFrame.fieldOutputs['U']
      fieldValues=displacement.values


# For each U value, print the nodeLabel
# and data members.

      for v in fieldValues:
          outline = "Node = %2.0d U[x] = %6.8f, U[y] = %6.8f, u[z] = %6.8f
" % (v.nodeLabel,
          v.data[0], v.data[1], v.data[2])
          file.write(outline)


##################
#UR  displacement rotation at nodes 1-6
```

```
#
#     displacementrotation=lastFrame.fieldOutputs['UR']
#     fieldValuesur=displacementrotation.values
#

# For each UR value, print the nodeLabel
# and data members.

#     for v in fieldValuesur:
#         outline = "Node = %2.0d UR[x] = %6.8f, UR[y] = %6.8f, uR[z] =
%6.8f  " % (v.nodeLabel,
#         v.data[0], v.data[1], v.data[2])
#         file.write(outline)

##################
#V  velocity at nodes 1-6

for i  in range(51):
#     timex = itertime*i
#     time = "Time = %6.6f" % (timex)
      file.write("\n")
#       file.write(time + "\n")
      lastFrame = odb.steps['Step-1'].frames[i]
      velocity=lastFrame.fieldOutputs['V']
      fieldValuesv=velocity.values


# For each V value, print the nodeLabel
# and data members.

      for v in fieldValuesv:
          outline = "Node = %2.0d V[x] = %6.8f, V[y] = %6.8f, V[z] = %6.8f
" % (v.nodeLabel,
          v.data[0], v.data[1], v.data[2])
          file.write(outline)


##################
#VR  velocity rotation at nodes 1-6

#     velocityrotation=lastFrame.fieldOutputs['VR']
#     fieldValuesvr=velocityrotation.values


# For each VR value, print the nodeLabel
# and data members.

#     for v in fieldValuesvr:
#         outline = "Node = %2.0d VR[x] = %6.8f, VR[y] = %6.8f, VR[z] =
%6.8f" % (v.nodeLabel,
#         v.data[0], v.data[1], v.data[2])
#         file.write(outline +"\n")
```

```
##################
#A  acceleration at nodes 1-6

#     acceleration=lastFrame.fieldOutputs['A']
#     fieldValuesa=acceleration.values


# For each A value, print the nodeLabel
# and data members.

#     for v in fieldValuesa:
#         outline = "Node = %2.0d A[x] = %6.8f, A[y] = %6.8f, A[z] =
%6.8f" % (v.nodeLabel,
#         v.data[0], v.data[1], v.data[2])
#         file.write(outline +"\n")


##################
#AR  acceleration rotation at nodes 1-6

#     accelerationrotation=lastFrame.fieldOutputs['AR']
#     fieldValuesar=accelerationrotation.values


# For each ar value, print the nodeLabel
# and data members.

#     for v in fieldValuesar:
#         outline = "Node = %2.0d AR[x] = %6.8f, AR[y] = %6.8f, AR[z] =
%6.8f" % (v.nodeLabel,
#         v.data[0], v.data[1], v.data[2])
#         file.write(outline +"\n")



file.close()

print "done! open results in excel"
```

Attachment D

U

```
Node =  1 U[x] = 0.00000000, U[y] = 0.00000000, u[z] = 0.00000000
Node =  2 U[x] = 0.00000000, U[y] = 0.00000000, u[z] = 0.00000000
Node =  3 U[x] = 0.00000000, U[y] = 0.00000000, u[z] = 0.00000000
Node =  4 U[x] = 0.00000000, U[y] = 0.00000000, u[z] = 0.00000000
Node =  5 U[x] = 0.00000000, U[y] = 0.00000000, u[z] = 0.00000000
Node =  6 U[x] = 0.00000000, U[y] = 0.00000000, u[z] = 0.00000000

Node =  1 U[x] = 0.00000000, U[y] = 0.00000000, u[z] = 0.00000000
Node =  2 U[x] = -0.00010812, U[y] = 0.01039741, u[z] = 0.00000000
Node =  3 U[x] = -0.00015341, U[y] = 0.01712761, u[z] = 0.00000000
Node =  4 U[x] = -0.00016988, U[y] = 0.02118577, u[z] = 0.00000000
Node =  5 U[x] = -0.00017518, U[y] = 0.02348901, u[z] = 0.00000000
Node =  6 U[x] = -0.00017691, U[y] = 0.02480394, u[z] = 0.00000000

Node =  1 U[x] = 0.00000000, U[y] = 0.00000000, u[z] = 0.00000000
Node =  2 U[x] = -0.00068229, U[y] = 0.02611179, u[z] = 0.00000000
Node =  3 U[x] = -0.00145377, U[y] = 0.05387673, u[z] = 0.00000000
Node =  4 U[x] = -0.00225930, U[y] = 0.08224734, u[z] = 0.00000000
Node =  5 U[x] = -0.00306724, U[y] = 0.11066009, u[z] = 0.00000000
Node =  6 U[x] = -0.00386590, U[y] = 0.13890955, u[z] = 0.00000000

Node =  1 U[x] = 0.00000000, U[y] = 0.00000000, u[z] = 0.00000000
Node =  2 U[x] = -0.00144777, U[y] = 0.03802200, u[z] = 0.00000000
Node =  3 U[x] = -0.00303150, U[y] = 0.07778664, u[z] = 0.00000000
Node =  4 U[x] = -0.00468169, U[y] = 0.11837572, u[z] = 0.00000000
Node =  5 U[x] = -0.00636329, U[y] = 0.15934844, u[z] = 0.00000000
Node =  6 U[x] = -0.00806546, U[y] = 0.20057075, u[z] = 0.00000000

Node =  1 U[x] = -0.00000000, U[y] = -0.00000000, u[z] = 0.00000000
Node =  2 U[x] = -0.00213202, U[y] = 0.04612460, u[z] = 0.00000000
Node =  3 U[x] = -0.00399993, U[y] = 0.08930360, u[z] = 0.00000000
Node =  4 U[x] = -0.00571695, U[y] = 0.13070494, u[z] = 0.00000000
Node =  5 U[x] = -0.00735686, U[y] = 0.17116749, u[z] = 0.00000000
Node =  6 U[x] = -0.00897422, U[y] = 0.21135138, u[z] = 0.00000000

Node =  1 U[x] = -0.00000000, U[y] = 0.00000000, u[z] = 0.00000000
Node =  2 U[x] = -0.00340399, U[y] = 0.05824433, u[z] = 0.00000000
Node =  3 U[x] = -0.00679785, U[y] = 0.11640234, u[z] = 0.00000000
Node =  4 U[x] = -0.01023630, U[y] = 0.17493969, u[z] = 0.00000000
Node =  5 U[x] = -0.01371552, U[y] = 0.23382194, u[z] = 0.00000000
Node =  6 U[x] = -0.01721262, U[y] = 0.29285473, u[z] = 0.00000000

Node =  1 U[x] = 0.00000000, U[y] = -0.00000000, u[z] = 0.00000000
Node =  2 U[x] = -0.00484826, U[y] = 0.06946047, u[z] = 0.00000000
Node =  3 U[x] = -0.01002000, U[y] = 0.14118902, u[z] = 0.00000000
Node =  4 U[x] = -0.01546026, U[y] = 0.21474628, u[z] = 0.00000000
Node =  5 U[x] = -0.02110750, U[y] = 0.28968191, u[z] = 0.00000000
Node =  6 U[x] = -0.02684813, U[y] = 0.36523107, u[z] = 0.00000000
```

Attachment E

```
Step-1
iteration  00
Node =  1 F[x] = 0.00000000, F[y] = 0.00000000, F[z] = 0.00000000
Node =  2 F[x] = 0.00000000, F[y] = 0.00000000, F[z] = 0.00000000
Node =  3 F[x] = 0.00000000, F[y] = 0.00000000, F[z] = 0.00000000
Node =  4 F[x] = 0.00000000, F[y] = 0.00000000, F[z] = 0.00000000
Node =  5 F[x] = 0.00000000, F[y] = 0.00000000, F[z] = 0.00000000
Node =  6 F[x] = 0.00000000, F[y] = 0.00000000, F[z] = 0.00000000

Step-1
iteration  01
Node =  1 F[x] = 0.00000000, F[y] = 0.00000000, F[z] = 0.00000000
Node =  2 F[x] = -0.00578963, F[y] = 0.08462580, F[z] = 0.00000004
Node =  3 F[x] = -0.00936150, F[y] = 0.15126581, F[z] = 0.00000089
Node =  4 F[x] = -0.01221042, F[y] = 0.21087037, F[z] = 0.00000190
Node =  5 F[x] = -0.01508493, F[y] = 0.27074319, F[z] = 0.00000251
Node =  6 F[x] = -0.00230485, F[y] = 0.03090382, F[z] = -0.00000002

Step-1
iteration  02
Node =  1 F[x] = 0.00000000, F[y] = 0.00000000, F[z] = 0.00000000
Node =  2 F[x] = -0.03595529, F[y] = 0.20869097, F[z] = -0.00000005
Node =  3 F[x] = -0.08157023, F[y] = 0.44302678, F[z] = 0.00000062
Node =  4 F[x] = -0.13097882, F[y] = 0.68658113, F[z] = 0.00000176
Node =  5 F[x] = -0.18097685, F[y] = 0.93152463, F[z] = 0.00000299
Node =  6 F[x] = -0.01060434, F[y] = 0.06562003, F[z] = -0.00000003

Step-1
iteration  03
Node =  1 F[x] = 0.00000000, F[y] = 0.00000000, F[z] = 0.00000000
Node =  2 F[x] = -0.05663102, F[y] = 0.25984818, F[z] = -0.00000018
Node =  3 F[x] = -0.10521249, F[y] = 0.50141752, F[z] = 0.00000002
Node =  4 F[x] = -0.15093702, F[y] = 0.73607385, F[z] = 0.00000036
Node =  5 F[x] = -0.19604990, F[y] = 0.96921706, F[z] = 0.00000077
Node =  6 F[x] = -0.02025223, F[y] = 0.08960050, F[z] = -0.00000005

Step-1
iteration  04
Node =  1 F[x] = 0.00000000, F[y] = 0.00000000, F[z] = 0.00000000
Node =  2 F[x] = -0.10797606, F[y] = 0.35114986, F[z] = 0.00000035
Node =  3 F[x] = -0.21880151, F[y] = 0.70646465, F[z] = 0.00000195
Node =  4 F[x] = -0.33073202, F[y] = 1.06337464, F[z] = 0.00000370
Node =  5 F[x] = -0.44326475, F[y] = 1.42114854, F[z] = 0.00000522
Node =  6 F[x] = -0.03539374, F[y] = 0.11616284, F[z] = 0.00000001

Step-1
iteration  05
Node =  1 F[x] = 0.00000000, F[y] = 0.00000000, F[z] = 0.00000000
Node =  2 F[x] = -0.15665475, F[y] = 0.41382641, F[z] = -0.00000019
Node =  3 F[x] = -0.32309613, F[y] = 0.83849019, F[z] = 0.00000045
Node =  4 F[x] = -0.49433100, F[y] = 1.26828098, F[z] = 0.00000164
Node =  5 F[x] = -0.66625112, F[y] = 1.69880235, F[z] = 0.00000298
```

```
Node =  6 F[x] = -0.05079001, F[y] = 0.13631809, F[z] = -0.00000005

Step-1
iteration  06
Node =  1 F[x] = 0.00000000, F[y] = 0.00000000, F[z] = 0.00000000
Node =  2 F[x] = -0.18102637, F[y] = 0.43981221, F[z] = 0.00000023
Node =  3 F[x] = -0.34614688, F[y] = 0.86303800, F[z] = 0.00000077
Node =  4 F[x] = -0.50506985, F[y] = 1.27944505, F[z] = 0.00000123
Node =  5 F[x] = -0.66198921, F[y] = 1.69360685, F[z] = 0.00000157
Node =  6 F[x] = -0.06296769, F[y] = 0.14923733, F[z] = 0.00000003

Step-1
iteration  07
Node =  1 F[x] = 0.00000000, F[y] = 0.00000000, F[z] = 0.00000000
Node =  2 F[x] = -0.24728975, F[y] = 0.49786538, F[z] = 0.00000001
Node =  3 F[x] = -0.51422358, F[y] = 1.01011312, F[z] = 0.00000109
Node =  4 F[x] = -0.78871250, F[y] = 1.52757406, F[z] = 0.00000263
Node =  5 F[x] = -1.06559992, F[y] = 2.04665661, F[z] = 0.00000398
Node =  6 F[x] = -0.07924187, F[y] = 0.16352290, F[z] = -0.00000003

Step-1
iteration  08
Node =  1 F[x] = 0.00000000, F[y] = 0.00000000, F[z] = 0.00000000
Node =  2 F[x] = -0.25614330, F[y] = 0.50454140, F[z] = -0.00000011
Node =  3 F[x] = -0.50578892, F[y] = 1.00426972, F[z] = 0.00000014
Node =  4 F[x] = -0.75385344, F[y] = 1.50281465, F[z] = 0.00000073
Node =  5 F[x] = -1.00248837, F[y] = 2.00178838, F[z] = 0.00000139
Node =  6 F[x] = -0.08656663, F[y] = 0.16904515, F[z] = -0.00000003

Step-1
iteration  09
Node =  1 F[x] = 0.00000000, F[y] = 0.00000000, F[z] = 0.00000000
Node =  2 F[x] = -0.27688622, F[y] = 0.51907343, F[z] = 0.00000024
Node =  3 F[x] = -0.54874700, F[y] = 1.03474307, F[z] = 0.00000076
Node =  4 F[x] = -0.81707168, F[y] = 1.54798102, F[z] = 0.00000086
Node =  5 F[x] = -1.08337140, F[y] = 2.05979323, F[z] = 0.00000076
Node =  6 F[x] = -0.09310102, F[y] = 0.17355049, F[z] = 0.00000002

Step-1
iteration  10
Node =  1 F[x] = 0.00000000, F[y] = 0.00000000, F[z] = 0.00000000
Node =  2 F[x] = -0.30236977, F[y] = 0.53521603, F[z] = -0.00000024
Node =  3 F[x] = -0.62530786, F[y] = 1.08235848, F[z] = 0.00000072
Node =  4 F[x] = -0.95780957, F[y] = 1.63469017, F[z] = 0.00000235
Node =  5 F[x] = -1.29336071, F[y] = 2.18862152, F[z] = 0.00000414
Node =  6 F[x] = -0.09761786, F[y] = 0.17645514, F[z] = -0.00000007

Step-1
iteration 11........
```